

```

1 #include <iostream>
2 #include <sstream>
3 #include <map>
4
5 #include <Kinect.h>
6 #include <opencv2¥opencv.hpp>
7
8 #include <atlbase.h>
9
10 // 次のように使います
11 // ERROR_CHECK( ::GetDefaultKinectSensor( &kinect ) );
12 // 書籍での解説のためにマクロにしています。実際には展開した形で使うことを検討してください。
13 #define ERROR_CHECK( ret ) ¥
14     if ( (ret) != S_OK ) { ¥
15         std::stringstream ss; ¥
16         ss << "failed " #ret " " << std::hex << ret << std::endl; ¥
17         throw std::runtime_error( ss.str().c_str() ); ¥
18     }
19
20 class KinectApp
21 {
22 private:
23
24     // Kinect SDK
25     CComPtr<IKinectSensor> kinect = nullptr;
26
27     CComPtr<IColorFrameReader> colorFrameReader = nullptr;
28
29     int colorWidth;
30     int colorHeight;
31     unsigned int colorBytesPerPixel;
32
33     ColorImageFormat colorFormat = ColorImageFormat::ColorImageFormat_Bgra;
34
35     // 表示部分
36     std::vector<BYTE> colorBuffer;
37
38 public:
39
40     KinectApp() デストラクタ
41     {
42         // Kinectの動作を終了する
43         if ( kinect != nullptr ) {
44             kinect->Close();
45         }
46     }
47
48     // 初期化
49     void initialize()
50     {
51         // デフォルトのKinectを取得する
52         ERROR_CHECK( ::GetDefaultKinectSensor( &kinect ) );
53         ERROR_CHECK( kinect->Open() );
54
55         // カラーリーダーを取得する
56         CComPtr<IColorFrameSource> colorFrameSource;
57         ERROR_CHECK( kinect->get_ColorFrameSource( &colorFrameSource ) );
58         ERROR_CHECK( colorFrameSource->OpenReader( &colorFrameReader ) );
59
60         // デフォルトのカラー画像のサイズを取得する
61         CComPtr<IFrameDescription> defaultColorFrameDescription;
62         ERROR_CHECK( colorFrameSource->get_FrameDescription( &defaultColorFrameDescription ) );
63         ERROR_CHECK( defaultColorFrameDescription->get_Width( &colorWidth ) );
64         ERROR_CHECK( defaultColorFrameDescription->get_Height( &colorHeight ) );
65         ERROR_CHECK( defaultColorFrameDescription->get_BytesPerPixel( &colorBytesPerPixel ) );
66         std::cout << "default : " << colorWidth << ", " << colorHeight << ", " <<
        colorBytesPerPixel << std::endl;
67
68         // カラー画像のサイズを取得する
69         CComPtr<IFrameDescription> colorFrameDescription;
70         ERROR_CHECK( colorFrameSource->CreateFrameDescription(
71             colorFormat, &colorFrameDescription ) );
72         ERROR_CHECK( colorFrameDescription->get_Width( &colorWidth ) );

```

```

73 ERROR_CHECK( colorFrameDescription->get_Height( &colorHeight ) );
74 ERROR_CHECK( colorFrameDescription->get_BytesPerPixel( &colorBytesPerPixel ) );
75 std::cout << "create : " << colorWidth << ", " << colorHeight << ", " <<
colorBytesPerPixel << std::endl;
76
77 // バッファを作成する
78 colorBuffer.resize( colorWidth * colorHeight * colorBytesPerPixel );
79 }

```

```

81 void run()
82 {
83     while ( 1 ) {
84         update();
85         draw();
86
87         auto key = cv::waitKey( 10 );
88         if ( key == 'q' ) {
89             break;
90         }
91     }
92 }

```

kadaiprog(); として課題を実装する。

94 private:

```

95 // データの更新処理
96 void update()
97 {
98     updateColorFrame();
99 }

```

```

100 // カラーフレームの更新
101 void updateColorFrame()
102 {
103     // フレームを取得する
104     CComPtr<IColorFrame> colorFrame;
105     auto ret = colorFrameReader->AcquireLatestFrame( &colorFrame );
106     if ( FAILED( ret ) ) {
107         return;
108     }
109
110     // 指定の形式でデータを取得する
111     ERROR_CHECK( colorFrame->CopyConvertedFrameDataToArray(
112         colorBuffer.size(), &colorBuffer[0], colorFormat ) );
113 }

```

```

114 // データの表示処理
115 void draw()
116 {
117     drawColorFrame();
118 }

```

```

119 // カラーデータの表示処理
120 void drawColorFrame()
121 {
122     #if 0
123     // カラーデータを表示する
124     cv::Mat colorImage( colorHeight, colorWidth, CV_8UC4, &colorBuffer[0] );
125     cv::imshow( "Color Image", colorImage );
126     #else
127     cv::Mat colorImage( colorHeight, colorWidth, CV_8UC4, &colorBuffer[0] );
128     cv::Mat harfImage;
129     cv::resize( colorImage, harfImage, cv::Size(), 0.5, 0.5 );
130     cv::imshow( "Harf Image", harfImage );
131     #endif
132 }

```

これを1にすればifブロックが実行される。

スパルミスは見苦しいので直す。

```

133 void main()
134 {
135     try {
136         KinectApp app;
137         app.initialize();
138         app.run();

```

クラス名

インスタンスを作る。その後クラス KinectApp の中のメンバーやメンバー関数が使えます。

app.メンバー  
app.メンバー関数 } のように。

void kadaiprog()

```
145     }  
146     catch ( std::exception& ex ) {  
147         std::cout << ex.what() << std::endl;  
148     }  
149 }  
150
```